Assignment 5: Data Visualization

Summer Heschong

Fall 2024

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Visualization

Directions

1. Rename this file `<FirstLast>_A05_DataVisualization.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure your code is tidy; use line breaks to ensure your code fits in the knitted output.
5. Be sure to **answer the questions** in this assignment document.
6. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up your session

1. Set up your session. Load the tidyverse, lubridate, here & cowplot packages, and verify your home directory. Read in the NTL-LTER processed data files for nutrients and chemistry/physics for Peter and Paul Lakes (use the tidy `NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv` version in the Processed_KEY folder) and the processed data file for the Niwot Ridge litter dataset (use the `NEON_NIWO_Litter_mass_trap_Processed.csv` version, again from the Processed_KEY folder).

2. Make sure R is reading dates as date format; if not change the format to date.

```r
#1
library(tidyverse);library(lubridate);library(here); library(cowplot)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
## here() starts at /home/guest/EDE_Fall2024
##
##
## Attaching package: 'cowplot'
##
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
getwd()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```
here()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```
PeterPaul.chem.nutrients <- read.csv(here(
  'Data/Processed_KEY/NTL-LTER_Lake_Chemistry_Nutrients_PeterPaul_Processed.csv'),
  stringsAsFactors = TRUE)
Litter <- read.csv(here(
  'Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_Processed.csv'),
  stringsAsFactors = TRUE)
#2
PeterPaul.chem.nutrients$sampledate <- ymd(PeterPaul.chem.nutrients$sampledate)
Litter$collectDate <- ymd(Litter$collectDate)
```

Define your theme

3. Build a theme and set it as your default theme. Customize the look of at least two of the following:

- Plot background
- Plot title
- Axis labels
- Axis ticks/gridlines
- Legend

```
#3
my_theme <-
  theme(
    panel.background = element_rect(fill = 'lightblue'),
    plot.title = element_text(
      hjust = 0.5,
      size = 16,
      face= 'bold',
      colour = 'black')
  )
```

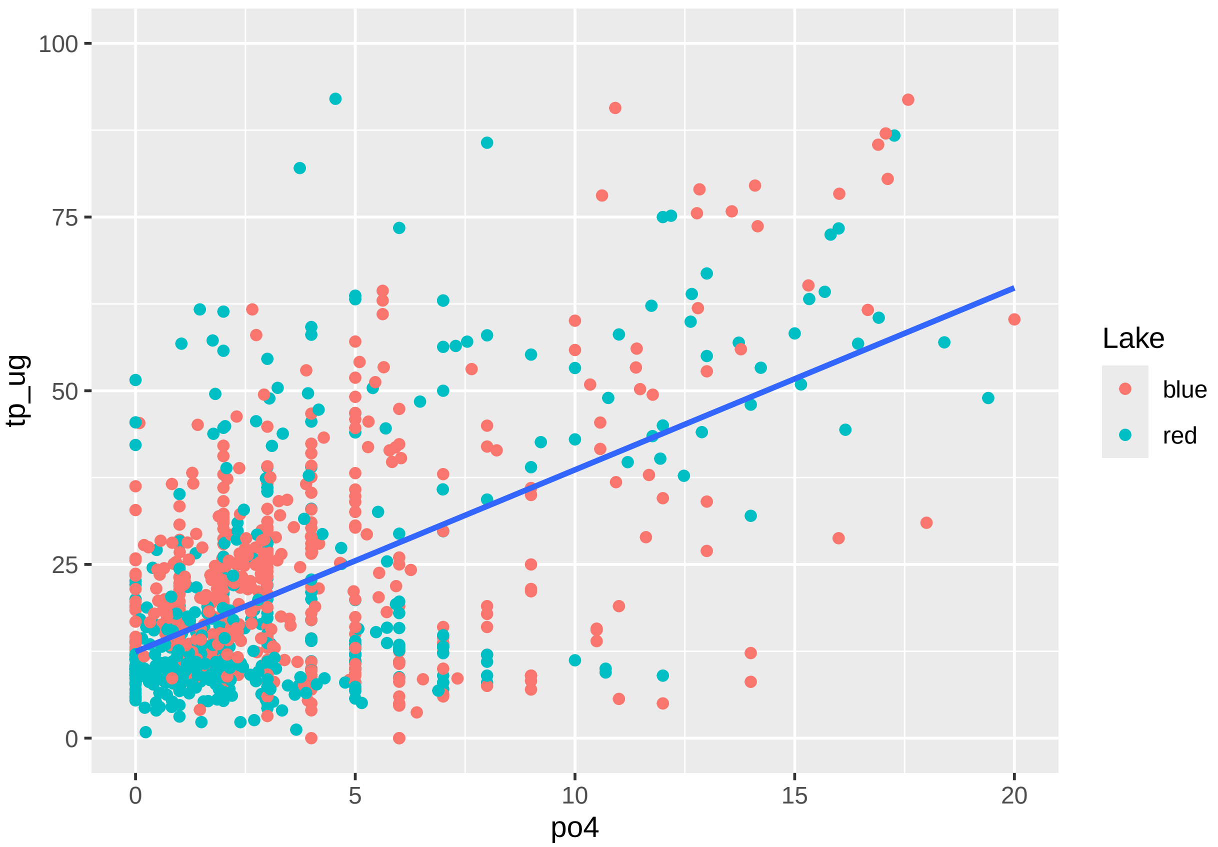
Create graphs

For numbers 4-7, create ggplot graphs and adjust aesthetics to follow best practices for data visualization. Ensure your theme, color palettes, axes, and additional aesthetics are edited accordingly.

4. [NTL-LTER] Plot total phosphorus (`tp_ug`) by phosphate (`po4`), with separate aesthetics for Peter and Paul lakes. Add line(s) of best fit using the `lm` method. Adjust your axes to hide extreme values (hint: change the limits using `xlim()` and/or `ylim()`).

```
#4
tpbypo4_plot <- PeterPaul.chem.nutrients %>%
  ggplot(aes(x=po4, y=tp_ug))+
  geom_point(aes(color=ifelse(lakename == 'Peter Lake', 'blue', 'red'))) +
  geom_smooth(method = 'lm',se=FALSE) +
  xlim(0, 20) +
  ylim(0, 100) +
  labs(color='Lake')  #I can't figure out how to change the labels
#of the blue and red dots in the legend
tpbypo4_plot
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 21986 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 21986 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



5. [NTL-LTER] Make three separate boxplots of (a) temperature, (b) TP, and (c) TN, with month as the x axis and lake as a color aesthetic. Then, create a cowplot that combines the three graphs. Make sure that only one legend is present and that graph axes are aligned.

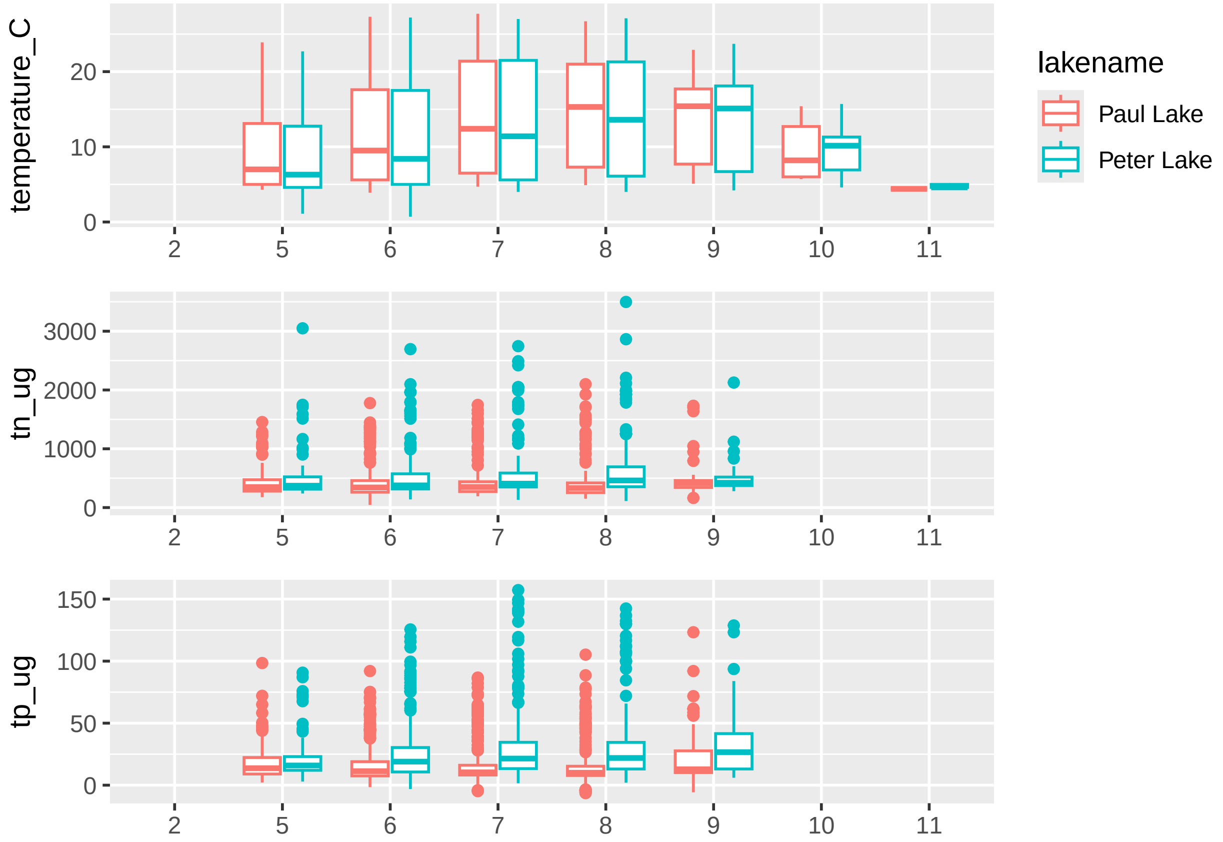
Tips: * Recall the discussion on factors in the lab section as it may be helpful here. * Setting an axis title in your theme to `element_blank()` removes the axis title (useful when multiple, aligned plots use the same axis values) * Setting a legend's position to "none" will remove the legend from a plot. * Individual plots can have different sizes when combined using `cowplot`.

```
#5
#convert 'month' to factor
PeterPaul.chem.nutrients$month <- factor(PeterPaul.chem.nutrients$month)

#Make boxplots for temp, tp, and tn, and remove x axis title and legend from all but one
Lake_boxplot_temp <- PeterPaul.chem.nutrients %>%
  ggplot(aes(x=month, y=temperature_C))+
  geom_boxplot(aes(color=lakename)) +
  theme(axis.title.x = element_blank())

Lake_boxplot_TP <- PeterPaul.chem.nutrients %>%
  ggplot(aes(x=month, y=tp_ug))+
  geom_boxplot(aes(color=lakename)) +
  theme(axis.title.x = element_blank(),
        legend.position = 'none')
```

```
Lake_boxplot_TN <- PeterPaul.chem.nutrients %>%
  ggplot(aes(x=month, y=tn_ug))+
  geom_boxplot(aes(color=lakename)) +
  theme(axis.title.x = element_blank(),
        legend.position = 'none')

#combine and align boxplots using cowplot
TpTnTemp_boxplot <- plot_grid(
  Lake_boxplot_temp,
  Lake_boxplot_TN,
  Lake_boxplot_TP,
  nrow=3, align='v', axis='x')
```

```
## Warning: Removed 3566 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).

## Warning: Removed 21583 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).

## Warning: Removed 20729 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

```
TpTnTemp_boxplot
```

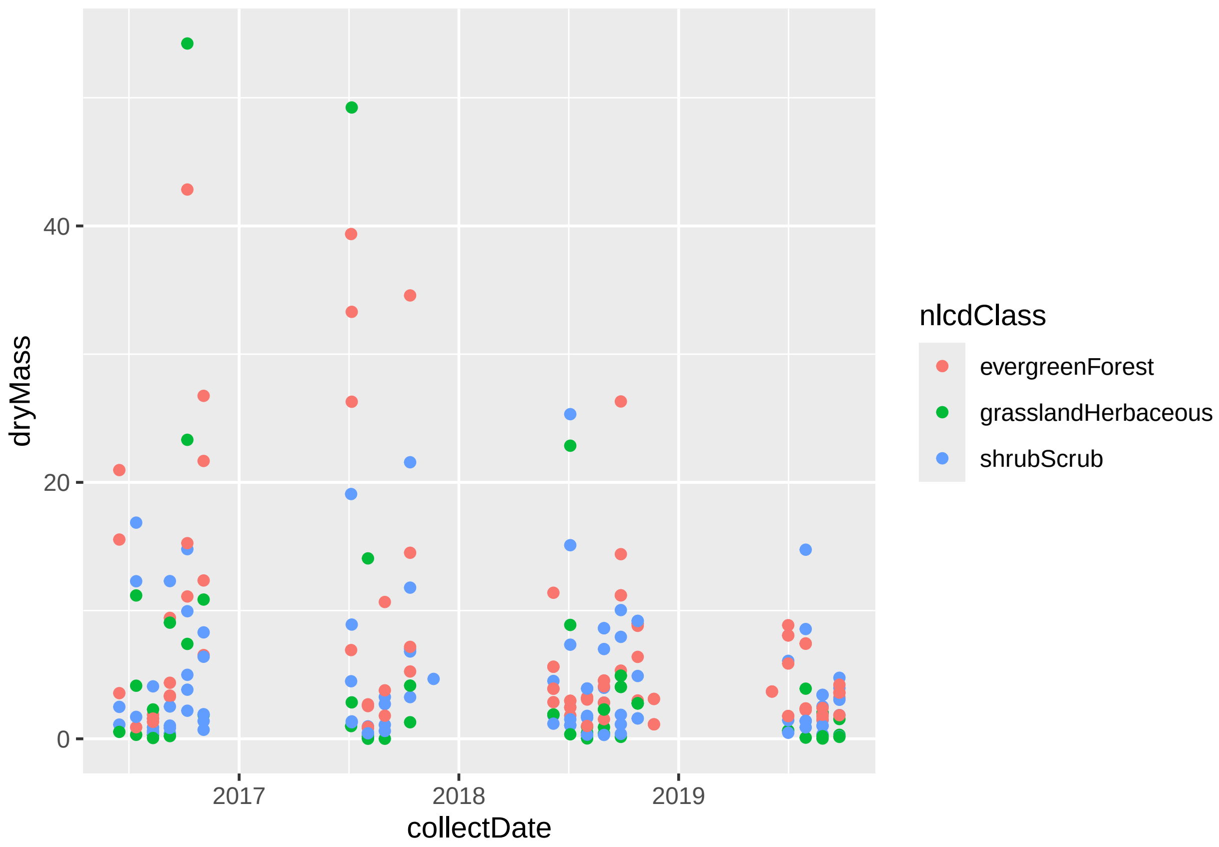


Question: What do you observe about the variables of interest over seasons and between lakes?

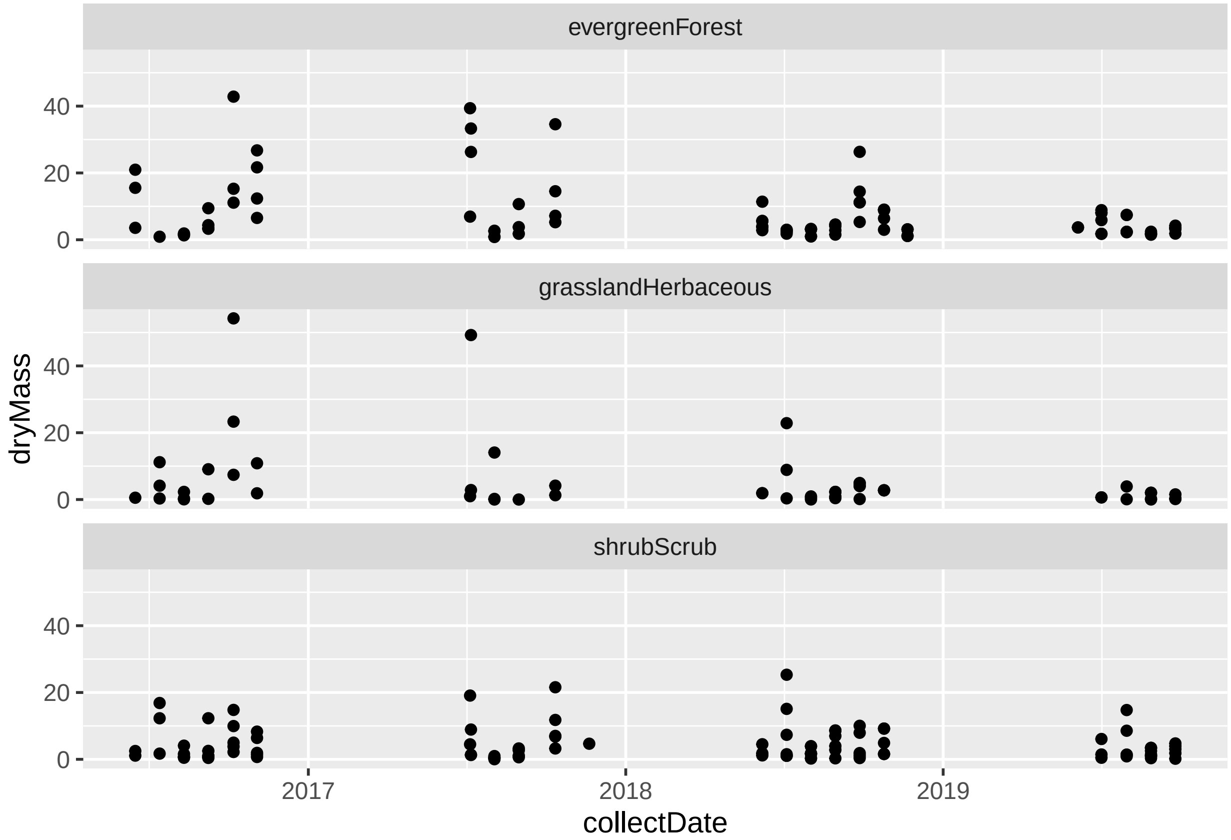
Answer: The temperature is highest in late summer for both of the lakes. Tn and Tp are slightly higher in Peter Lake than Paul Lake.

6. [Niwot Ridge] Plot a subset of the litter dataset by displaying only the "Needles" functional group. Plot the dry mass of needle litter by date and separate by NLCD class with a color aesthetic. (no need to adjust the name of each land use)
7. [Niwot Ridge] Now, plot the same plot but with NLCD classes separated into three facets rather than separated by color.

```r
#6
litter_plotcolor <- Litter %>%
  filter(functionalGroup=='Needles') %>%
  ggplot(aes(x=collectDate, y=dryMass, color=nlcdClass)) +
  geom_point()
litter_plotcolor
```



```r
#7
litter_plotfacet <- Litter %>%
  filter(functionalGroup=='Needles') %>%
  ggplot(aes(x=collectDate, y=dryMass)) +
  geom_point() +
  facet_wrap(vars(nlcdClass), nrow=3)
litter_plotfacet
```



Question: Which of these plots (6 vs. 7) do you think is more effective, and why?

Answer:I think that plot 7 is more effective because in plot 6 the points overlap and hide each other so it's hard to see all the data.